

Portability Considerations for FPGA-Based Implementations in an SCA Compliant System

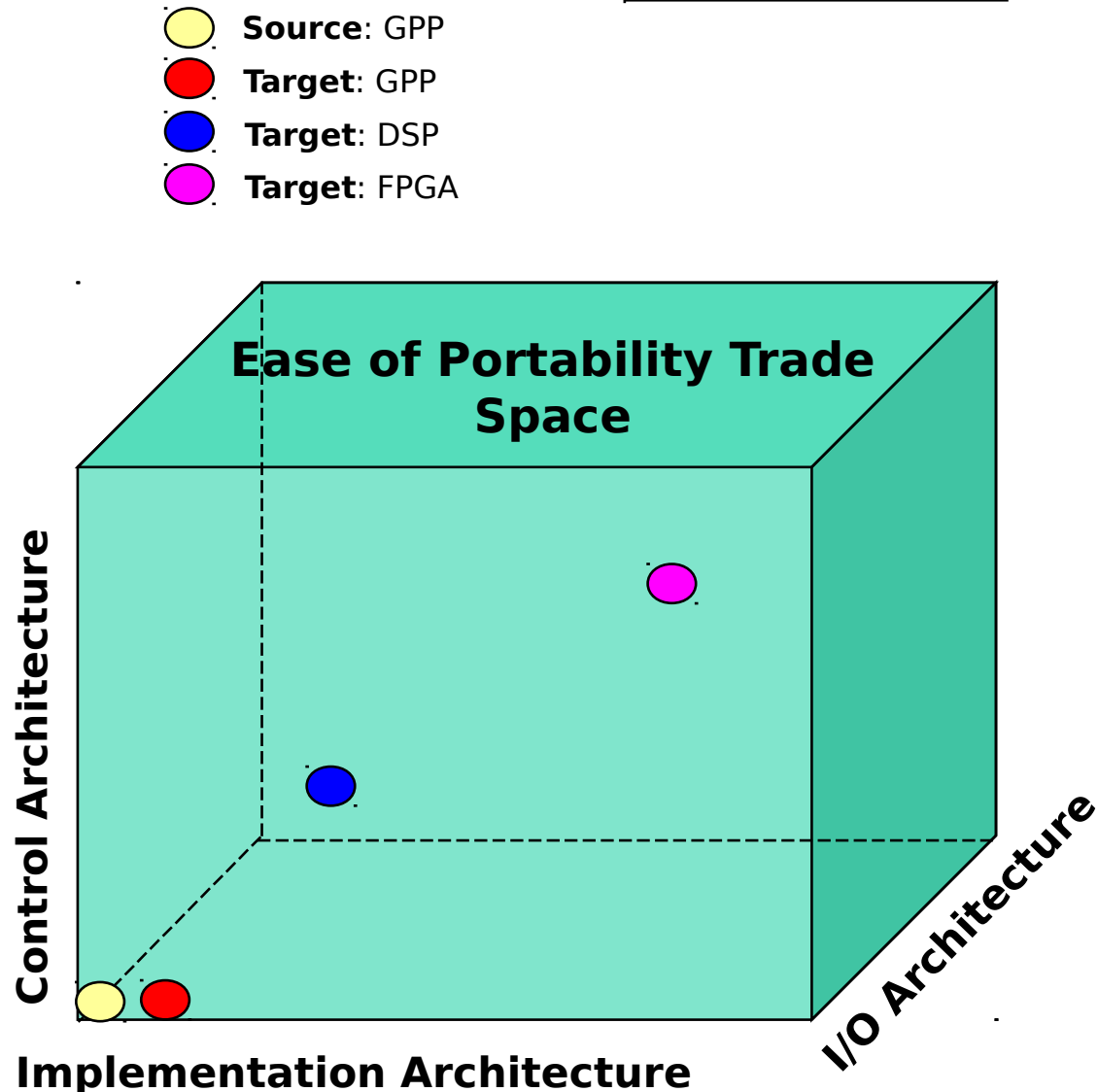
Vincent Kovarik Jr., Ph.D.

Harris Corporation

(321) 984-5631

vkovarik@harris.com

- **Portability does not imply reuse without modification.**
- **Portability is relative to the differences between (at least) three architectural dimensions of the source and target environments:**
 - Implementation
 - I/O
 - Control
- **Portability is enhanced by:**
 - Minimizing the distance between the source and target environments, or
 - Providing a common model of the hardware and software components, or
 - Generating implementations from high level models, or
 - All of the above.
- **Timing constraints run through all dimensions.**



- **Implementation**

- What is the fundamental implementation of the processing subsystem?
- The processor architecture directly affects the implementation language, approaches, and performance.

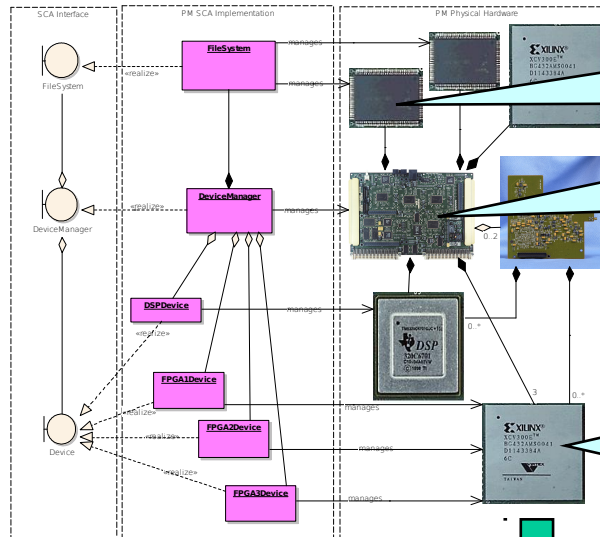
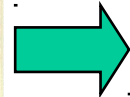
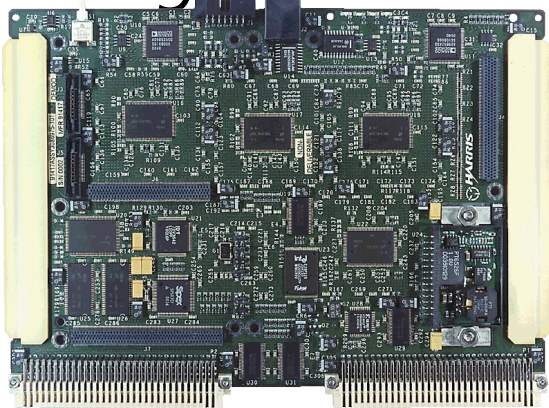
- **I/O**

- What are the I/O interfaces supported by the processing subsystem?
- Each processor type has different I/O mechanisms with differing levels of standardization.
- The I/O architecture is further complicated by multiple hardware designs incorporating multiple processors. This results in additional complexity of inter- and intra-board communications.

- **Control**

- What is the method of controlling the application on a given implementation architecture?
- Specification of an GPP-style API may not work for an FPGA-based processing system.

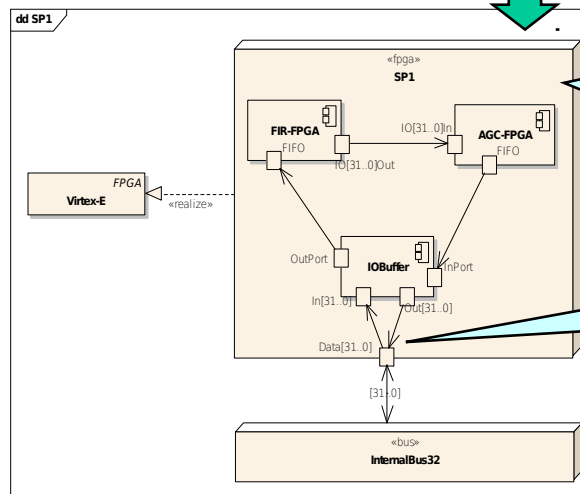
FPGA-Based, SCA Compliant Programmable Modem



Manage on-board flash memory as an SCA File System. Allows external access into the flash by 3rd party software.

Base card is represented as an SCA Device Manager. This maps to the use of the File System for flash and multiple devices on the card.

DSP and FPGAs are represented as discrete SCA Devices providing a fine-grained management and control capability.



Device components modeled in UML with software components mapped to deployment targets.

I/O connectivity identified within model.

- **SCA Device and Device Manager interfaces capabilities include:**
 - Device Manager interface for the board
 - Logical Device interface for each FPGA and the DSP
 - SCA File System interface for on-board flash
- **Demonstrated SCA Capabilities**
 - TCDL 10 Mb link with legacy TCDL terminal
 - Control of web cam and receipt of live video over link
 - Shutdown of TCDL waveform, load, and start of 274 Mb CDL waveform in under 10 seconds.
- **Currently being utilized on multiple programs including AEHF/NMT**

Specifying FPGA SCA Properties



Device Properties are defined using SCA Properties XML. This provides access to set/get property values from applications and external components.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE properties SYSTEM "properties.2.2.dtd">
<properties>
```

```
  <simple id="BoardID"
        type="long"
        name="BoardID"
        mode="readonly">
```

```
    <description>
```

```
      Board ID and revision, typically in date format (yyyymmdd).
```

```
    </description>
```

```
    <kind kindtype="configure"/>
```

```
  </simple>
```

```
  <simple id="DDS1TuningWord"
        type="double"
        name="DDS1TuningWord"
        mode="readwrite">
```

```
    <description>
```

```
      32-bit tuning word calculated by the formula, FTW = (Desired output freq x 2^N)/SYSCLK
```

```
    </description>
```

```
    <kind kindtype="configure"/>
```

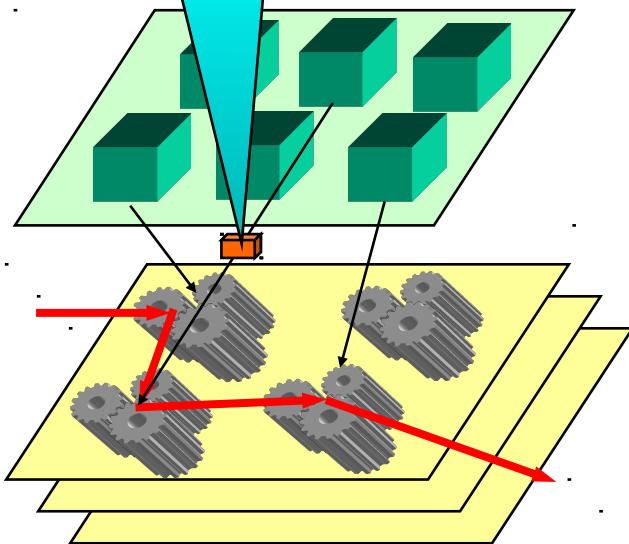
```
  </simple>
```

However, how those properties are mapped to underlying device characteristics is implemented in device-specific, non-portable code. Changes in the load characteristics requires re-writing software.

Mapping FPGA Implementations



Representing knowledge about the load format, location, register map, etc. provides a transformational mapping between the high-level SCA Property definition and the actual implementation.



```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE fpgaregisters SYSTEM "fpgaregisters.dtd">
<fpgaregisters id="PMControlFPGARegisterMap">

  <description>
    Harris Programmable Modem Control FPGA Register Map
  </description>

  <item id="BoardID"
        block="0"
        register="1"
        startbit="31"
        stopbit="0"
        type="long"
        access="read">
    <description>
      Board ID and revision, typically in date format (yyyymmdd).
    </description>
  </item>

  <item id="DDS1TuningWord"
        block="0"
        register="2"
        startbit="31"
        stopbit="0"
        type="double"
        access="readwrite">
    <description>
      32-bit tuning word calculated by the formula, FTW = (Desired output freq x
      2^N)/SYSCLK.
    </description>
  </item>
```

- **Understand the Problem**
 - Understanding the problem is **not** the same as understanding there is a problem.
- **There is no Panacea**
 - Progress towards portability requires continuing efforts throughout the community across a multitude of disciplines.
- **Integrated Design**
 - Developing a software radio requires collaborative participation of all contributors, e.g. hardware (digital and analog), waveform design, software implementation.